Signal, Image and Video Processing
https://doi.org/10.1007/s11760-023-02837-z

ORIGINAL PAPER

# Freshness uniformity measurement network based on multi-layer feature fusion and histogram layer

Ying Zang[1] · Chunan Yu[1] · Chenglong Fu[1] · Zhenfeng Xue[2] · Qingshan Liu[1] · Yong Zhang[1,3]

Received: 11 February 2023 / Revised: 8 October 2023 / Accepted: 13 October 2023


**Abstract**
The arrangement of products on supermarket freshness shelves exhibits a certain pattern and displays distinct texture characteristics. In recent years, many studies have applied texture extraction algorithms in deep learning, such as the Histogram Layer Residual Network (HistNet). However, this algorithm still has obvious disadvantages, such as neglecting the optimal representation of multi-scale texture features and lacking feature selection during extraction. To address these issues, this paper introduces a novel texture classification network—Multi-Scale Feature Histogram Network (MFHisNet). First, we design a Multi-Scale Feature Fusion Module (MF-Block) to achieve a multi-level representation of texture information. Then, we utilize an attention module (CBAM) to weight crucial information and suppress background interference for deeper level texture features. Experimental results demonstrate that the model achieves accuracies of 82.12 ±2.04%, 73.13±1.10%, and 83.46±0.62% on the GTOS-mobile, DTD, and MINC-2500 datasets, respectively. Furthermore, based on the proposed model, we propose a measurement method that uses cosine similarity to measure the uniformity of freshness placement, and the effectiveness of this method was verified on the dataset we collected.

**Keywords** Freshness uniformity measure · Texture recognition · Histogram layer · Multi-scale feature fusion · Attention mechanism

## 1 Introduction

The uniformity of the merchandise in the supermarket can not only improve the utilization of the space, but more importantly, it can increase customers' desire to buy. As shown in Fig. 1, the left image is more orderly than the right one, so the customer's purchase desire is stronger. Whether the goods are placed neatly or not can be represented by texture features.

Texture recognition is an important research area in computer vision with a wide range of applications, including material classification, terrain recognition, and medical imaging [1–5]. Classical texture recognition methods usually use a set of filters to convert texture images into local features, and then aggregate them into a global representation, such as texture histogram and bag-of-words methods [6–10]. In recent years, Deep Convolutional Neural Network (DCNN) has made significant progress in texture recognition, and researchers use deep learning to achieve automatic feature extraction to enhance feature representation [11–13]. Although deep learning has achieved great success in texture recognition, some researchers have proved theoretically and practically that using traditional features in texture recognition is often superior to some deep learning methods [14–17]. In order to integrate the advantages of traditional features, Peeples et al. [18] proposed a texture classification network - Histogram Layer for ResNet (HistRes), which integrates traditional histogram features into a deep learning framework.

HistNet used histogram layers to characterize the distribution of feature maps extracted by CNN models, providing additional texture information and significantly improving the accuracy of texture classification. However, textures in the natural world often exhibit scale variations, especially in supermarket. Although HistNet adds histogram layers at different positions in the backbone network to extract deeper-level features, but not fully aware of the texture char-

✉ Yong Zhang
zhyong@zjhu.edu.cn

1 School of Information Engineering, Huzhou University, Huzhou 313000, China

2 Huzhou Institute of Zhejiang University, Huzhou 313000, China

3 School of Computer and Information Technology, Liaoning Normal University, Dalian 116081, China

Published online: 19 November 2023

**Fig. 1** Commodities with different placement standards

acteristics of a variety of scales. In addition, the interference of background noise in the real scene is also a problem.

To address the aforementioned issues, this paper introduces a texture classification network based on histogram layers, multi-scale feature fusion, and attention mechanisms. Additionally, a new measure of freshness scene uniformity based on MFHisNet is designed.

The main contributions of this paper are as follows:

(1) We propose a novel texture classification network—Multi-Scale Feature Histogram Network (MFHisNet), and achieve competitive results on GTOS-mobile, DTD, and MINC-2500 three datasets.
(2) To extract multi-scale texture features, we introduce a generalized MF-Block module for capturing multi-scale features. We use histogram layers to model features at different scales, enriching texture information. To mitigate background noise interference, we incorporate attention mechanisms to enhance the importance of key features while suppressing background information.
(3) Based on the MFHisNet, we present a freshness uniformity measurement method based on cosine similarity. The feature extraction module of MFHisNet extracts features from standard and test images, calculates their cosine similarity to determine the similarity of textures, thereby assessing the uniformity of the test image.

## 2 Related work

This section reviews the related works, inculding the texture classification algorithm and the attention mechanism.

### 2.1 Texture classification algorithm

Texture recognition is one of the most important and challenging problems in computer vision and pattern recognition. In recent years, with the development of deep learning, convolutional neural networks have become an important tool for texture recognition. Cimpoi et al. [11] proposed a Fisher vector-based CNN classifier (FV-CNN), which is considered a milestone in texture classification and significantly improved the performance of texture classification. FV-CNN computes FV pooling on deep generic features such as deep convolution activation features. A disadvantage of the FV-CNN architecture is that the CNN feature extraction, texture encoding and classifier are learned separately. To jointly train in an end-to-end manner, Zhang et al. [19] proposed a deep texture encoding network (DeepTEN), building dictionary learning and feature pooling on a CNN architecture, which learns an unordered representation and provides good performance in material classification. But textures or materials do not always exhibit a disordered state, so local spatial information is still very necessary. To address this problem, Xue et al. [4] proposed a deep encoding pooling network (DEP) for ground terrain recognition, which integrates disordered texture details and local spatial information. The shortcoming of this architecture is that the features from different layers of the CNN are not fully utilized. In order to make full use of the features of different layers, Hu et al. [12] proposed Multi-level Texture Encoding and Representation (MuLTER), which can simultaneously extract low-level and high-level CNN features, and realizes the multi-level representation of texture features, while maintaining the texture details are preserved while the local spatial information is

preserved. Existing methods ignore visual texture properties, which are important features for describing real textured images, resulting in incomplete descriptions and inaccurate identifications. To address this issue, Zhai et al. [20] proposed a novel deep multi-attribute perception network (MAP-Net) to gradually learn visual texture attributes in a mutually reinforcing manner. A multi-branch network structure is designed to learn cascading global contexts by introducing similarity constraints at each branch, and an attribute transfer scheme to guide the spatial feature encoding of the next branch. More recently, Yang et al. [21] proposed DFAEN (Double-order Knowledge Fusion and Attentional Encoding Network), which takes advantage of attention mechanisms to aggregate first and second-order information for encoding texture features. Bu et al. [22] proposed Locality-aware coding advocates exploiting appropriately convolutional layer activations to constitute a powerful descriptor for texture classification under an end-to-end learning framework.

Some literatures show that traditional features outperform or are comparable to deep learning methods in texture classification (e.g., Basu et al. [14], Basu et al. [23]). In order to integrate the advantages of traditional features, Peeples et al. proposed a texture classification network HistNet that integrates histogram layers. However, HistNet has many obvious shortcomings. For example, HistNet ignores the multi-scale characteristics of texture, and the texture information of a single scale is not enough to achieve optimization. In addition, it ignores the complex environmental characteristics under real conditions, and does not screen the output features. According to these problems, we propose a more applicable texture classification network MFHisNet, which achieves better performance than HistNet.

### 2.2 Attention mechanism

Humans can naturally capture salient regions when observing objects. Based on this feature, attention mechanisms are introduced into computer vision tasks. The attention mechanism is a dynamic weight adjustment process for features. The attention mechanism has achi-eved great success in many vision tasks, including image classification, object detection, semantic segmentation, etc. This section analyzes and summarizes the commonly used attention modules.

In Xu et al. [24], a visual attention method is first proposed to model the importance of features in the image caption task. Then many methods start to focus on the attention mechanism. A residual attention network [25] is proposed with a spatial attention mechanism using downsampling and upsampling.upsampling. Hu et al. [26] first proposed the concept of channel attention and proposed SENet. The core of SENet is to propose a squeeze-and-excitation (SE) block to collect global information, capture inter-channel relationships, and improve representation capabilities. Since the SE block is a lightweight module, it is more conducive to the porting of the module. Besides, Luo et al. [27] proposed the attention module Shift-and-Balance Attention (SB). Compared with the SE attention module, the SB attention module uses Tanh as the activation function and is scaled by the learned control factor $\lambda$. However, SE and SB modules are only considered at the channel level, and the importance of different regions in the same channel may also be different. What's more, Woo et al. [28] proposed the convolutional block attention module (CBAM), which combines channel attention and spatial attention. The advantage of this module is that the channel attention and spatial attention are concatenated. CBAM places more emphasis on useful channels and strengthens the spatial attention areas of information. Due to its lightweight design, CBAM can be seamlessly integrated into any CNN architecture with negligible additional cost. Motivated by CBAM, GSoP [29] introduces a second-order pooling method for downsampling. NonLocal [30] proposes to build a dense spatial feature map. AANet [31] proposes to embed the attention map with position information into the feature. SkNet [32] introduces a selective channel aggregation and attention mechanism, and ResNeSt [33] proposes a similar split attention method. Due to the complicated attention operation, these methods are relatively large. To improve efficiency, GCNet [34] proposes to use a simple spatial attention module and replace the original spatial downsampling process. ECANet [35] introduces one-dimensional convolution layers to reduce the redundancy of fully connected layers and obtains more efficient results. Finally, Our proposed MFHisNet introduces the CBAM, which jointly introduces channel attention and spatial attention into the network. By changing the weights, it effectively increases the influence of key channel and key spatial features and suppresses useless features to eliminate background interference, thereby significantly improving the performance.

## 3 Proposed method

Based on the background of freshness texture recognition in complex scenes such as supermarkets, our met-hod proposes a high-performance texture recognition network MFHisNet. The innovation of this network is mainly reflected in three aspects: a multi-scale feature fusion module based on histogram layers MF-Block, attention mechanism, and freshness uniformity measure based on MFHisNet.

The overall structure of the network is shown in Fig. 2. First, the input image uses ResNet as the backbone network for deep feature extraction (GTOS-mobile uses Resnet18, DTD and MINC-2500 use Resnet50). Secondly, two branches are designed to select the features of Block3 and Block4 for feature processing. The first branch is the features output by Block4 go through the Global Average Pooling (GAP), BN

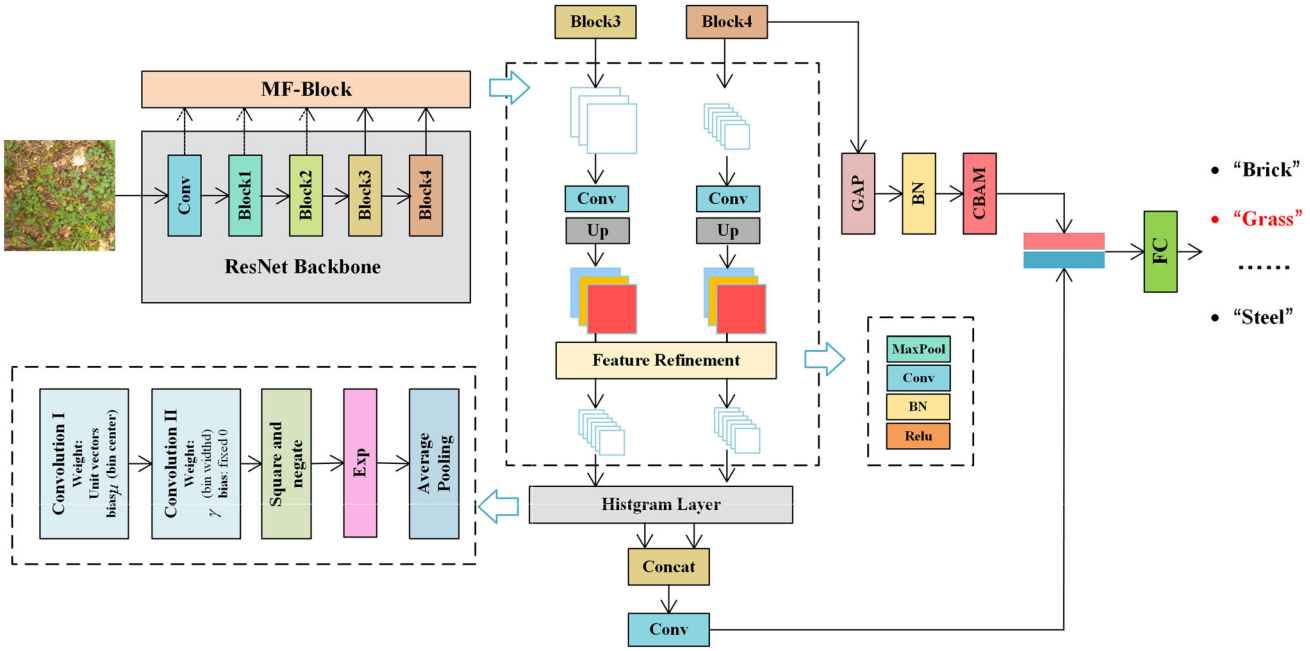



**Fig. 2** Overall structure of MFHisNet

and CBAM. The CBAM can filter useful features and solve the problem of severe background information interference by increasing the influence of key channels and key spatial features. The second branch is MF-Block, its input is the features output by Resnet's Block3 and Block4. After the features of different levels are refined, the histogram layer is used to model the feature distribution to directly obtain the texture information. MF-Block can effectively integrate information at different levels, increase the diversity and effectiveness of texture information. Finally, the information of the two branches is fused in the final stage to achieve the final classification task.

The backbone of our network uses two versions of Resnet, namely Resnet18 and Resnet50. Block3 of Resn-et18 consists of two Blocks, each Block consists of two 3×3×256 convolutions, the input size is 14×14, and the output size is 7×7. Bolck4 is also composed of two Blocks, each Block consists of two 3×3×516 convolutions, the input size is 7×7, and the output size is 1×1. Compared with Resnet18, Resnet50 has more network layers. Its Block3 is composed of six Blocks, and each Block is composed of three convolutions of 1×1×256, 3×3×256, and 1×1×1024. The input and output are the same as Resnet18. Bolck4 is also composed of three Block blocks, and each block is composed of 1×1×512, 3×3×512, and 1×1×2048 convolutions.

### 3.1 MF-block module

Due to the scale variability of textures, single-level features are insufficient in effectively representing textures. To ensure comprehensive representation of different textures and achieve finer classification, we propose a generalized module known as MF-Block. This module takes Block3 and Block4 of ResNet as inputs. Firstly, it preprocesses the input features through channel normalization and scale normalization. Subsequently, it utilizes a feature refinement module to further process the extracted features, eliminating redundant information. Finally, the processed features are output. The MF-Block module enhances the diversity and effectiveness of texture information, leading to a significant improvement in texture classification accuracy.

**Scale normalization** As shown in Fig. 2, let $F_t \in R^{C_t \times H_t \times W_t} (t = 1, 2, ..., 5)$ denote the feature vector output by the $t$-th layer, and $F_t$ has different spatial scales $H_t \times W_t$ and different number of channels $C_t$ on different layers $t$. Experience has proved that the fusion of shallow features in deep features can improve the classification effect. Based on a large number of experiments, $F_4$ and $F_5$ are selected as the input of MF-Block. In order to facilitate subsequent processing, the input features are first channel-normalized to convert the tensors $F_4$ and $F_5$ to $C_4$ channels, which is achieved by using $1 \times 1$ convolution. Then it is upsampled by bilinear interpolation to a fixed dimension $H_4 \times H_5$.

**Feature refinement** Histogram layer is an effective method for aggregating information, which can integrate the advantages of handcrafted features and deep learning to maximize the performance of texture classification. Before the normalized features are sent to the histogram layer, they go through a max pooling layer with a stride of 2 and a convolution kernel of $3 \times 3$, a $1 \times 1$ convolutional layer with a

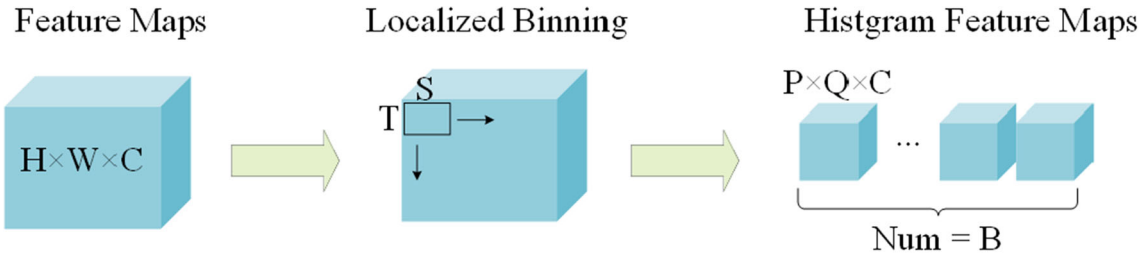



**Fig. 3** Histogram layer operation process

stride of 1, a Batch Normalization layer and the ReLu layer perform feature refinement, as shown in Fig. 2. Compared with only using $1 \times 1$ convolution with stride 2, this method can reduce the loss of information. It can be seen from the following experiments that this method greatly improves the accuracy of classification. The refined features are denoted as $V_t \in R^{C_5 \times H_5 \times W_5} (t = 4, 5)$.

**Histogram layer** The refined features $V_4$ and $V_5$ are, respectively, passed through the histogram layer to capture the texture information in the image, and the result can be expressed as $H_t \in R^{C \times H \times W} (t = 4, 5)$. Where $C$ is equal to $C_5/4$, the histogram layer uses a $2 \times 2$ sliding window, and the number of bins the number is 16, and the output feature map size is $2 \times 2$. After $H_4$ and $H_5$ are fused, they are used as supplementary information, and the features of $F_5$ after GAP, BN and CBAM are further fused as final features, which can greatly improve the performance of texture classification.

### 3.2 Implementation details of histogram layers

Histogram layer is a generalized module proposed by HistNet, which can be embedded into other network structures for end-to-end training. Since supermarket freshness contains rich texture information, we introduce histogram layer in MFHisNet. Histogram layers can directly capture the texture features of images by characterizing the distribution of feature maps extracted from CNN models. Through concat neural network and histogram information, combine the advantages of handcrafted features and deep learning to maximize the performance of texture analysis.

The operation process of the histogram layer is shown in Fig. 3. Input a feature map of $H \times W \times C$, use a sliding window of size $S \times T$ to calculate the normalized frequency count and assign it to $B$ bins, and finally get a of size $B$ with size $P \times Q \times C$ histogram layer feature map.

The histogram layer uses RBFs to model the histogram. RBFs use a Gaussian function as the kernel function, which provides a smoother approximation to the histogram. The closer the eigenvalue is to the center of the bin, the closer the value of RBFs will be to 1, the farther away from the center of the bin, the closer the value of RBFs will be to 0. Furthermore, RBFs are more robust to small changes in the center and width of bins than standard histogram operations due to the smoothness and soft-boxing mechanism of RBFs. The mean value of the RBFs (denoted as $\mu_{bc}$, where $b \in \{1, 2, ..., B\}$, $c \in \{1, 2, ..., C\}$) as the center of the $b$-th bin on the $c$ channel. The bandwidth of the RBFs (denoted as $\gamma_{bc}$) is taken as the width of the $b$-th bin on the $c$ channel. The binning operation of the histogram values uses a sliding window of $S \times T$ to compute normalized frequency counts (denoted as $Y_{pqbc}$), which are stored in row $p$ and column $q$ of the output of the histogram layer. The normalized frequency is calculated as follows:

$$g = x_{p+s,q+t,c} - \mu_{bc} \tag{1}$$

$$Y_{pqbc} = \frac{1}{ST} \sum_{s=1}^{S} \sum_{t=1}^{T} e^{-\gamma_{bc}^2 g^2} \tag{2}$$

where $x_{p+s,q+t,c}$ represents the feature value of the corresponding position in the feature map.

The histogram layer supports end-to-end learning by back-propagating to update the center and width of the bins. The gradient of the parameters $\mu_{bc}$ and $\gamma_{bc}$ of the histogram layer with sliding window size $S \times T$ are calculated as follows:

$$\frac{\partial Y_{pqbc}}{\partial \mu_{bc}} = \frac{2}{ST} \sum_{s=1}^{S} \sum_{t=1}^{T} e^{-\gamma_{bc}^2 g^2} \times \gamma_{bc}^2 \times g \tag{3}$$

$$\frac{\partial Y_{pqbc}}{\partial \mu_{bc}} = \frac{-2}{ST} \sum_{s=1}^{S} \sum_{t=1}^{T} e^{-\gamma_{bc}^2 g^2} \times \gamma_{bc} \times g^2 \tag{4}$$

where the values of $\mu_{bc}$ and $\gamma_{bc}$ are continuously updated using a gradient descent method within the histogram layer, which involves updating the centers and widths of the bins.

In Eqs. (3) and (4), the gradient values are functions of the feature map values and the distance from the bin center. If the feature map value is farther from the bin center, the corresponding gradient value will also be smaller. Conversely, if the feature map value is closer to the bin center, the gradient value will be larger.

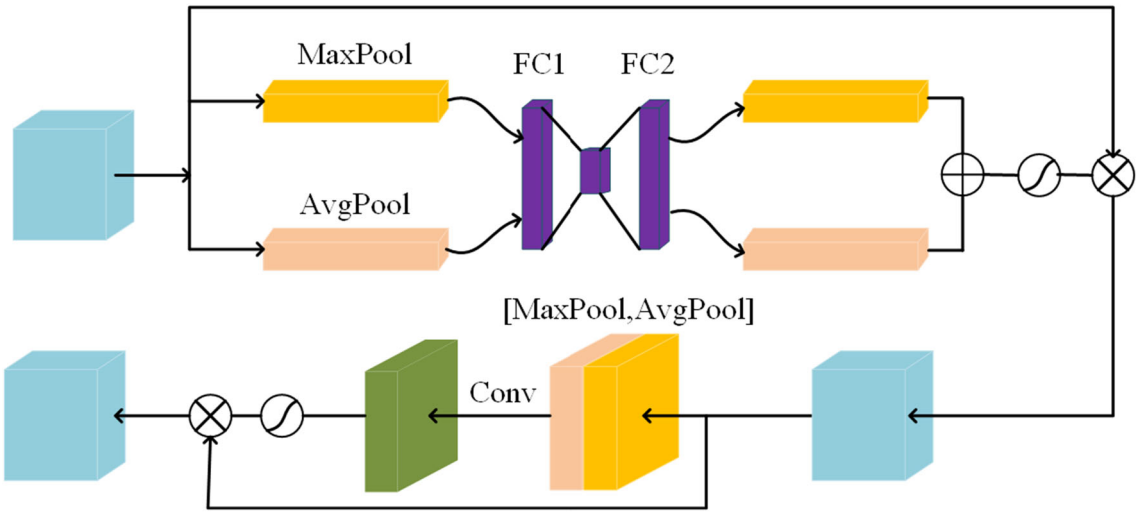



**Fig. 4** CBAM

The structure of the histogram is shown in Fig. 2. The convolution of $1 \times 1 \times c$ is used on the input feature map to reduce the number of channels of the input feature, where $C$ represents the new channel dimension of the feature map. After reducing the channel dimension, first assign each eigenvalue to each bin center (i.e., minus $\mu_{bc}$). Then use the convolution of $1 \times 1 \times b$ to calculate the feature center of each bin for each feature map. Among them, the weights in the convolution kernel are all fixed to 1, and each bias is used as a learnable bin center. After feature values have been assigned to bins, the centered features are multiplied by the bandwidth ($\gamma_{bc}$) to combine the feature distributions for each bin. The width of each bin is then calculated by using convolution of $1 \times 1 \times B$ on each feature map, taking the weights as learnable bin widths, and fixing the bias to 0. The gradient contribution to each bin is calculated by the RBF activation function in Eq. (2). The gradient contribution of each feature to each bin is between 0 and 1. Feature contributions from local spatial regions are then computed via average pooling to compute feature-normalized frequency counts for each bin.

### 3.3 Feature screening

Due to the presence of a significant amount of background information in complex environments, it can interfere with the accuracy of texture recognition. To address this issue and enhance the model's generality, we have introduced the CBAM attention module. This module weights crucial information and suppressing interference from background information. Additionally, we have incorporated an extra preprocessing step for our collected freshness data to filter out some noise before it enters the network.

CBAM consists of two sub-modules of channel attention and spatial attention. The channel attention module takes the input feature map $F$ through global max pooling and global average pooling, respectively, and obtains two $1 \times 1 \times C$ feature maps. Then the obtained feature maps are sent to a two-layer neural network (MLP), and then the features output by the MLP are subjected to an addition operation based on bitwise addition, and after the sigmoid activation operation, the final channel attention feature, namely $M_c$. Finally, the $M_c$ and the input feature map $F$ are subjected to a bitwise multiplication operation to generate the input features needed in the next step.

The specific calculation process of channel attention is as follows:

$$A(x) = AvgPool(x) \tag{5}$$

$$M(x) = MaxPool(x) \tag{6}$$

$$M_c(F) = \sigma(MLP(A(F) + M(F)) \tag{7}$$

where A(x) and M(x) are the outputs of AvgPoll and MaxPool, and $\sigma$ is the sigmoid activation function.

The spatial attention module takes the feature map output by the channel attention module as input. First, global average pooling and global max pooling are performed based on the channel, and then the channel splicing operation is performed. Then, after a $7 \times 7$ convolution operation, the dimension is reduced to 1 channel. Then generate the spatial attention feature through sigmoid, namely $M_s$. Finally, multiply the feature with the input feature of the module to get the final generated feature.

The spatial attention calculation process is as follows:

$$M_s(F) = \sigma(f^{7\times7}([A(F^{'}); M(F^{'})]) \tag{8}$$

The overall structure of CBAM is shown Fig. 4.

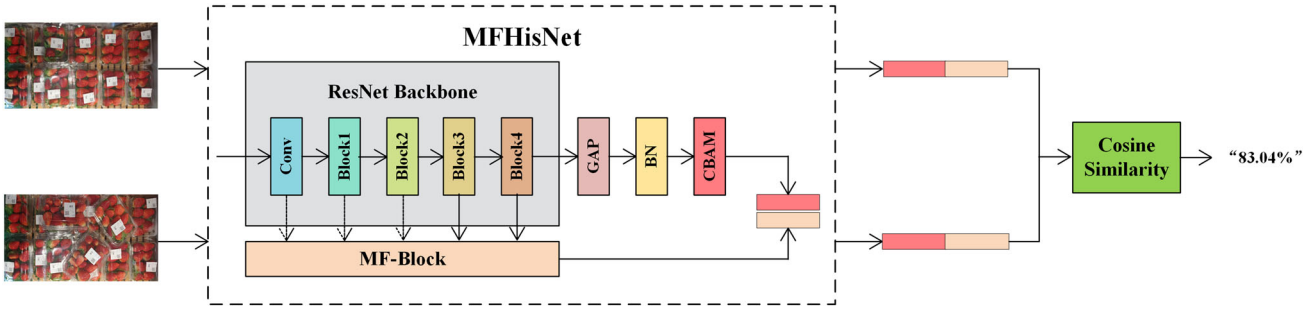



**Fig. 5** Flowchart of the uniformity detection of freshness in supermarkets

The whole calculation process of CBAM is as follows:

$$F'' = M_s((M_c(F) \otimes F)) \otimes (M_c(F) \otimes F) \tag{9}$$

### 3.4 Method for measuring the uniformity of freshness

In order to assess the uniformity of freshness product placement in supermarkets, we have introduced a supermarket freshness placement uniformity measurement method based on cosine similarity. This method utilizes the MFHisNet network to map image data into feature vectors in a high-dimensional space. Subsequently, it evaluates similarity through vector measurement methods such as cosine similarity. Finally, it utilizes a threshold-setting approach to further assess the uniformity of the products placed within the camera frame, as illustrated in Fig. 5.

First, we begin by inputting two similar freshness product images (standard image and test image). Next, we proceed to the image preprocessing stage. Due to uncertainties in the image acquisition process, scenarios with excessive background or small targets may appear. Therefore, it is necessary to perform operations such as image segmentation to extract scenes containing only freshness items, thereby increasing the number of useful features for subsequent processing. The role of the preprocessing stage is to enhance image quality, emphasize the image information of interest for subsequent processing, and suppress noise or unnecessary information. Following that, enter the feature extraction stage. Processed images are individually fed into MFHisNet for feature extraction, and the extracted features are then output. In this context, we treat the MFHisNet model as a black box, focusing solely on its input and output.

Finally, uniformity analysis. We utilize a similarity calculation method by comparing the feature vectors extracted by MFHisNet to measure the difference between the standard image and the test image. Cosine similarity is a commonly used vector similarity metric. In subsequent experiments, we also compared other methods such as Euclidean distance and Manhattan distance. By calculating cosine similarity, a score between 0% and 100% is obtained. The closer it is to 100%, the more similar the arrangement of the two images. Conversely, a greater difference indicates a more irregular state. In practical applications, the uniformity of freshness products can be determined by setting a threshold.

## 4 Experimental results

In this section, we first introduce the dataset and implementation details used for the experiments, and then conduct an exhaustive ablation study to verify the effectiveness of our proposed algorithm. We then compare our proposed MFHisNet method with other state-of-the-art texture classification algorithms. Finally, we use the dataset collected by ourselves to verify the effectiveness of our proposed supermarket freshness measurement algorithm.

### 4.1 Datasets and experimental details

**GTOS-mobile** GTOS-mobile contains 31 categories, such as grass, wood chips, cement and steel with a total of 37381 images. These images are not from the existing GTOS dataset, but from hand-held mobile phone videos of similar terrain. The dataset contains images of different resolutions, and only 256×256 images are used in this work. For the GTOS-mobile dataset using ResNet18 as the backbone network, there is only one separate training and testing split, but 5 different initialization experiments are performed.

**DTD** [36] DTD studies the problem of texture description and aims to identify descriptive texture properties. This dataset is different from some datasets that consider the identification of materials. Although the properties that can be described are related to materials, the properties do not represent the materials themselves. For example, the same texture properties may apply to trees and ground, and the ground can also contain dent, layered and other texture properties. That is, descriptive texture properties depend on human judgment, supporting human-centric tasks. The dataset consists of 5640 images and contains 47 categories, each of which is


**Fig. 6** Partial labels for three datasets

described by one or more words from the vocabulary, such as ribbon, bubble, knit, dimpled.

**MINC-2500** MINC-2500 is a subset of the MINC dataset intended to identify the material of each pixel in an image. Contains 23 categories, such as wood, sky, stone and water with a total of 54,625 images, of which the sample size is $362 \times 362$. Since each category is uniformly sampled, no original resolution images are required. ResNet50 is used as the backbone network for both DTD and MINC-2500 datasets, and published train and test splits are used in each experiment. The visualization pictures of some labels of the three datasets are shown in Fig. 6.

**Experiment settings** This paper adopts a training method similar to [4, 12, 18]. First resize the input image to $256 \times 256$, then randomly crop the image from 80% to 100% with a random aspect ratio of 3/4 to 4/3, then center crop. Image size became 224 × 224, in addition, data augmentation was performed by random horizontal flipping ($p = 0.5$), and images were normalized by subtracting the mean of each channel and dividing by the standard deviation of each channel. The training settings for each network are as follows: the batch size is 64, and the cross-entropy loss function is used. SGD with momentum (0.9), the learning rate decays by a factor of 1 every 10 epochs, and training stops after 30 epochs. The initial learning rates for the newly added and pretrained layers are 0.01 and 0.001, respectively.

## 4.2 Ablation study

### 4.2.1 MF-block module input feature layer selection

Due to the scale variability of textures, MF-Block can achieve multi-level representation of texture features, which not only captures more texture details but also retains local spatial information, increasing the diversity and effectiveness of texture features. However, the choice of which layers can achieve the best effect needs to be explained by experiments, so we conduct ablation experiments on the input of MF-Block. The ablation experiment starts from the high-level semantic layer of the backbone network and gradually accumulates forward. The experimental results are shown in Table 1.

It can be seen from the experimental data in the table that the effect of selecting Block3 and Block4 of the backbone network is the best, and the highest accuracy is achieved on the three datasets.

### 4.2.2 Feature refinement layer

The feature refinement layer is used to further refine the normalized features, filter useful features, and reduce unnecessary redundancy. Traditional methods are generally implemented by a series of $1 \times 1$ convolutions with a stride of 2. We have optimized this method, using the combination of MaxPool and Conv, MaxPool can be regarded as "hard sampling,"

**Table 1** MF-block module selects different input feature layers

| Layers | GTOS-mobile (%) | DTD (%) | MINC-2500 (%) |
|---|---|---|---|
| Block4 | 79.22 ± 0.77 | 71.60 ± 0.90 | 82.42 ± 0.33 |
| Block3-4 | **80.18 ± 1.14** | **71.81 ± 0.97** | **82.81 ± 0.46** |
| Block2-4 | 79.26 ± 0.81 | 71.69 ± 0.77 | 82.72 ± 0.58 |
| Block1-4 | 79.57 ± 1.69 | 71.48 ± 0.74 | 82.63 ± 0.43 |
| Conv-Block4 | 79.05 ± 1.10 | 71.39 ± 0.97 | 82.79 ± 0.50 |

**Table 2** Accuracy ablation experiment of feature refinement layer

| Methods | GTOS-mobile (%) | DTD (%) | MINC-2500 (%) |
|---|---|---|---|
| | 80.45 ± 1.51 | 71.54 ± 0.91 | 82.93 ± 0.59 |
| ✓ | **82.12 ± 2.04** | **73.13 ± 1.10** | **83.46 ± 0.62** |

**Table 3** Accuracy ablation experiments of MFHisNet network on three datasets

| CBAM | MF block | GTOS-mobile (%) | DTD (%) | MINC-2500 (%) |
|---|---|---|---|---|
| | | 79.22 ± 0.77 | 71.60 ± 0.90 | 82.42 ± 0.33 |
| ✓ | | 80.00 ± 0.82 | 72.36 ± 1.09 | 83.07 ± 0.57 |
| | ✓ | 80.18 ± 1.41 | 71.81 ± 0.97 | 82.81 ± 0.46 |
| ✓ | ✓ | **82.12 ± 2.04** | **73.13 ± 1.10** | **83.46 ± 0.62** |

Conv can be regarded as "soft sampling", MaxPool selects the element with the highest activation degree, which helps for classification, Conv can be a smooth transition between elements, which helps not to lose spatial information. Compared with the traditional method, this method can effectively reduce the loss of information, thereby improving the classification accuracy. The experimental results are shown in Table 2.

It can be seen from the table that the feature refinement module plays a very critical role and greatly improves the accuracy of texture classification.

#### 4.2.3 Propose ablation experiments of each module of the network

MFHisNet mainly contains two core modules: MF-Block module and attention module. To verify the role of each module in the network, we conduct ablation experiments on three datasets. The experimental results are shown in Table 3.

As shown in Table 3, on the GTOS-mobile, DTD, and MINC-2500 datasets, the addition of the attention mechanism alone improved the average accuracy by 0.78%, 0.76%, and 0.65%, respectively. Adding the MF-Block module alone improved the average accuracy by 0.96%, 0.21%, and 0.39%, respectively. The final architecture achieved an average accuracy improvement of 2.90%, 1.53%, and 1.04% on the three datasets, demonstrating the effectiveness of our approach. Additionally, performance was even better on the GTOS-mobile dataset, which is because the data in GTOS-mobile is captured in the most authentic natural environment using handheld smartphones, aligning well with the original design of our model framework.

The model size of our method has two types. One is the model on the GTOS-mobile dataset with trainable parameters of 11.41M. The other is a model on DTD and the MINC-2500 dataset with a trainable parameter of 23.81M. The two differences are that the former uses resnet18 and the latter uses resnet50.

### 4.3 Comparison with the state-of-the-art

To validate the effectiveness of the algorithm, we compared our algorithm with state-of-the-art algorithms on GTOS-mobile, DTD, and MINC-2500 three datasets. The comparison results are shown in Table 4.

As shown in Table 5, on the three datasets of GTOS-mobile, DTD and MINC-2500, the average accuracy of MFHisNet is 2.37%, 1.15% and 1.04% higher than HistNet, respectively.

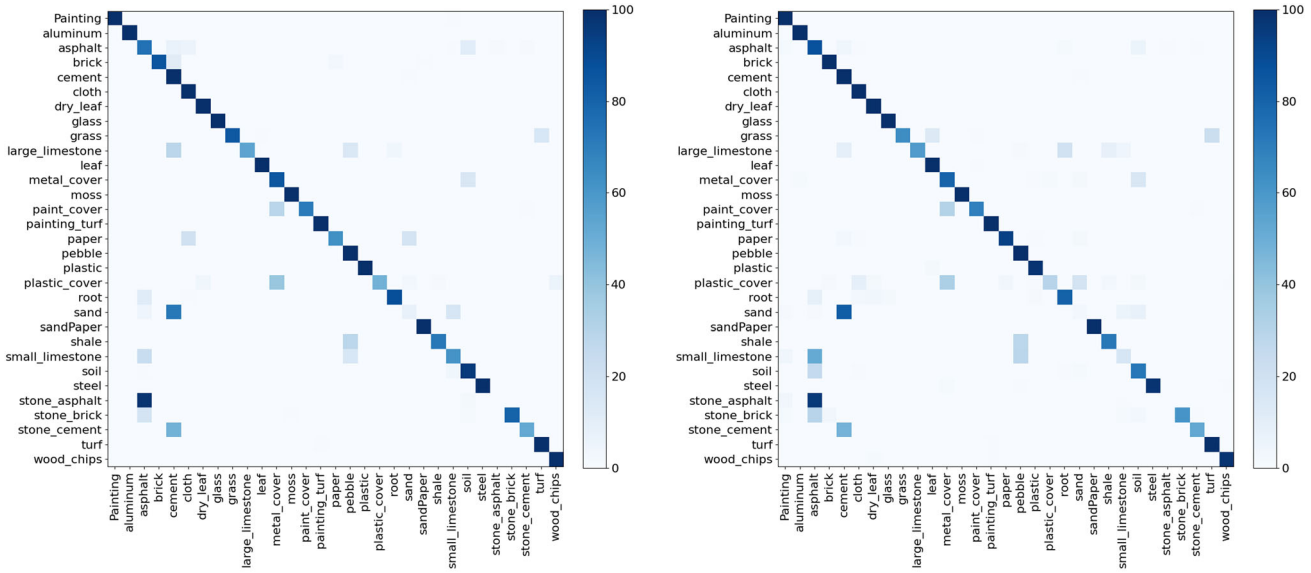
We show the confusion matrix of the two models MFHisNet and HistNet on the GTOS-mobile dataset in the form of a heat map. As shown in Fig. 7. We judge that our model is better than HistNet by comparing the depth of the squares in the figure. For example, the square on the diagonal represents the number of samples that the model predicts correctly, that is, the correct rate of the model. The larger the value on the diagonal, the darker the grid and the better the performance of the model. The values on the off-diagonal indicate the number of samples that the model predicted wrongly, that is, the error rate of the model. The smaller the value on the

**Table 4** Comparison experiment of accuracy between MFHisNet network and other methods

| Methods | GTOS-mobile (%) | DTD (%) | MINC-2500 (%) |
|---|---|---|---|
| GAP | 70.82 | – | – |
| GAP* | 76.09 ± 0.91 | 73.07 ± 0.79 | 83.01 ± 0.38 |
| FV-CNN | – | 72.30 | 63.10 |
| DeepTEN | 74.20 | 69.60 | 80.40 |
| DEP | 76.07 | 73.20 | 82.00 |
| MuLTER | 78.20 | – | 82.20 |
| Locality-aware coding | – | 71.10 | – |
| DFAEN | – | 73.20 | 81.66 |
| HistNet | 79.75 ± 0.84 | 71.98 ± 1.23 | 82.42 ± 0.33 |
| MFHisNet | **82.12 ± 2.04** | **73.13 ± 1.10** | **83.46 ± 0.62** |

**Table 5** Uniformity measurement experiment

| Standard image | Test image | Euclidean distance | Cosine distance (%) | Manhattan distance |
|---|---|---|---|---|
| Standard 1 | Test 1.1 | 15.37 | 85.24 | 244 |
| Standard 1 | Test 1.2 | 16.51 | 83.04 | 271 |
| Standard 2 | Test 2.1 | 13.90 | 89.75 | 213 |
| Standard 2 | Test 2.2 | 15.02 | 87.91 | 224 |




(a) MFHisNet (b) HistNet

**Fig. 7** Confusion matrix comparison between MFHisNet and HistNet

off-diagonal, the lighter the color of the square, and the better the performance of the model. In these two ways, it can be seen from the figure that our model is better than HistNet.

## 4.4 Freshness uniformity measurement

In order to verify the feasibility of our method in the real scene, we conducted a large number of experiments on the data in the real scene. Our experiment is designed as follows:

(1) Take strawberries and oranges as examples, select a standard image and two tests, respectively. The pictures use different vector distance measurement methods, respectively. The closer the Euclidean distance and the Manhattan distance are to 0, and the clos er the cosine similarity is to 100%, the more tidy the placement is. The experimental data and results are shown in Fig. 8 and Table 5, respectively.

Experiments have shown that when the number of clutter is increased, the neatness will also decrease. And our method

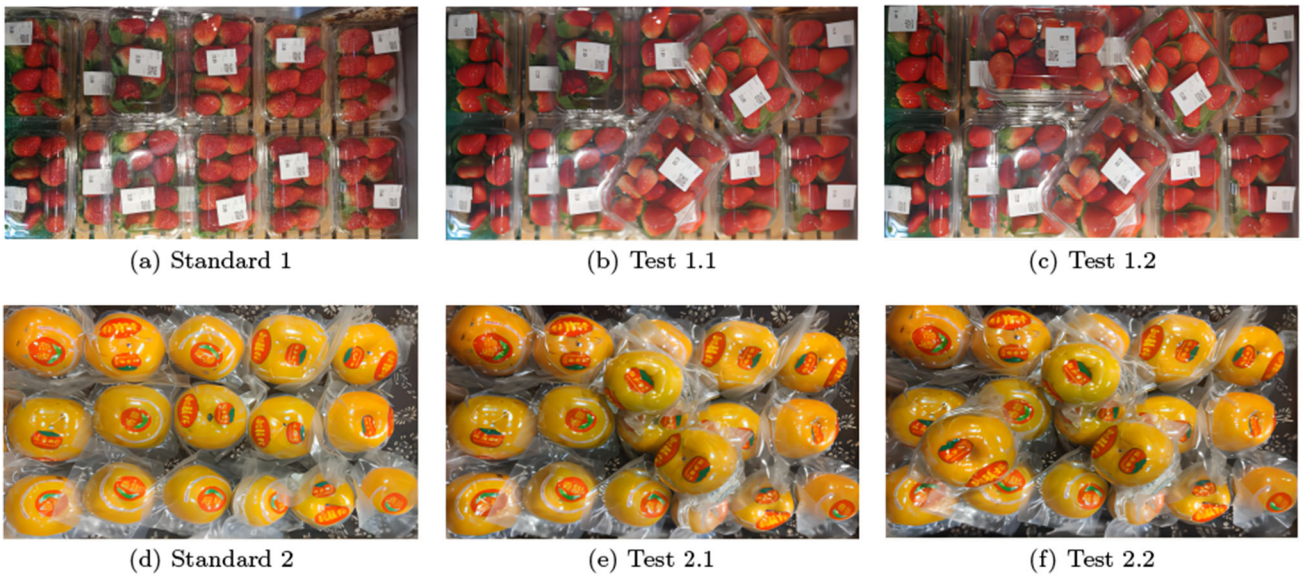



**Fig. 8** Data for experiment 1

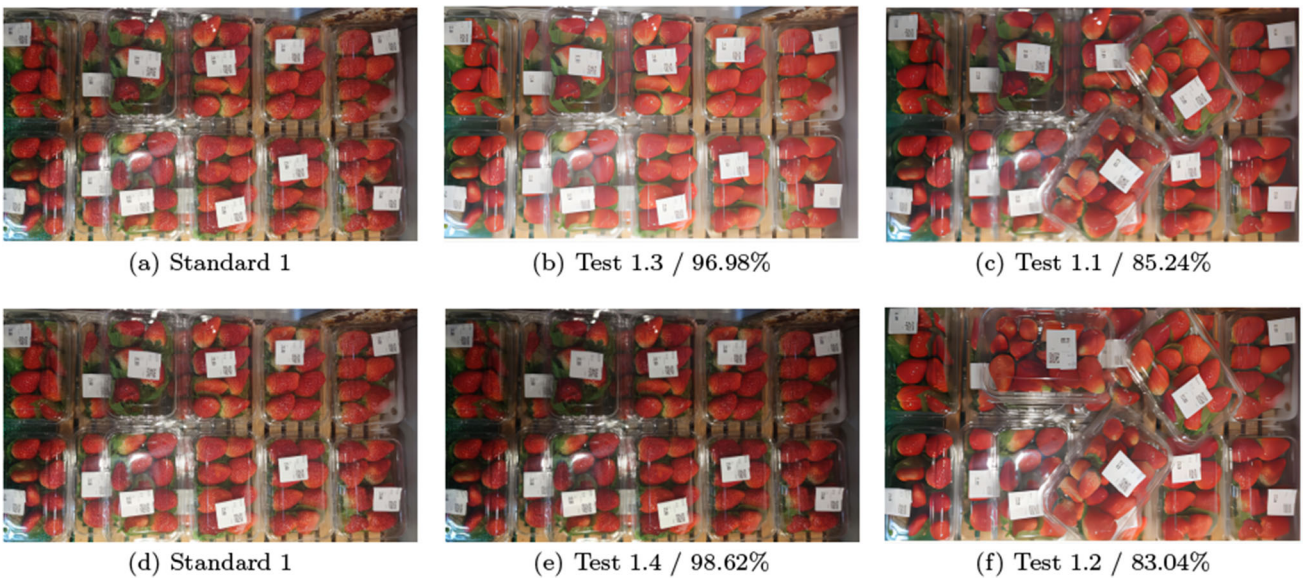



**Fig. 9** Light and dark experiment

also has good discrimination for subtle differences, proving the effectiveness of our method.

(2) In order to verify the influence of environmental factors on the uniformity measurement, we collected data under both bright and dark environmental conditions and compared them with cluttered scenes. Since cosine similarity is more intuitive, we only use cosine distance for similarity measurement in subsequent experiments. The experimental results are shown in Fig. 9.

The experimental results show that the light–dark change is clearly distinguishable from the cluttered arrangement, which proves that our method has good robustness to the light–dark change.

(3) In the real scene, for the data captured by handheld mobile devices, the shooting distance is not fixed. In order to verify the robustness of our method to the shooting distance, the following experiments are carried out. The experimental results are shown in Fig. 10.

The experimental results show that the shooting distance is clearly distinguishable from cluttered placement, which proves that our method has good robustness to the shooting distance.


**Fig. 10** Shooting distance experiment

**Fig. 11** Different angles experiment

(4) In real scenes, for data collected by handheld mobile devices, the shooting angle is not fixed. To verify the robustness of our method to camera angles, the following experiments are performed. The experimental results are shown in Fig. 11.

Below is a comparison of different angles to prove that our model has no effect on the shooting angle factor. We analyze the comparison method, as shown in Fig. 11. We compare the two pictures (a) and (b), and we can see that the two items are placed at different degrees but at the same angle, and the similarity is low; we compare the two pictures (a) and (c), and the items. The degree of confusion is the same as (b) but the angle is different, and the resulting similarity is close to (b). The conclusion can be obtained: the model is related to the degree of confusion of commodities, and has nothing to do with the angle.

## 5 Conclusions

In this paper, we propose a novel and more general texture classification network, MFHisNet, based on HistNet. Firstly, MF-Block module is proposed to realize multilevel representation of texture features, increase the diversity and effectiveness of texture information, this addresses the issue of insufficient expression of texture features and significantly improves the accuracy of texture classification. Secondly, the CBAM attention mechanism is introduced to filter irrelevant information, enhancing the influence of key information and reducing interference from noise like background, making it more suitable for complex real-world environments. After that, compared with the advanced methods, our method has significant improvement on DTD, GTOS-mobile and MINC-2500 datasets, which fully proves the effectiveness of our method in texture recognition and other tasks. Finally, based on MFHisNet model, a measure method of supermarket freshness uniformity based on cosine similarity is designed, which achieves excellent performance in practical scenarios.

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China under Grant 61772252, the Scientific Research Foundation of the Education Department of Liaoning Province under Grant LJKZ0965, and the Huzhou Science and Technology Plan Project under Grants 2022GZ08 and 2023ZD2004.

**Author Contributions** YZ and YZconceived the idea. YZ and CY realized the idea and wrote the main manuscript text. CF and CY prepared all figures and tables. YZ, ZX, and QL provided supervision. All authors reviewed the manuscript.

**Data availability** The GTOS-mobile dataset can be accessed via https://drive.google.com/file/d/1Hd1G7aKhsPPMbNrk4zHNJAzoXvUzWJ9M/view. The DTD dataset can be accessed via https://www.robots.ox.ac.uk/vgg/data/dtd/ and MINC-2500 dataset can be accessed via http://opensurfaces.cs.cornell.edu/static/minc/minc-2500.tar.gz.

## Declarations

**Conflict of interest** The authors declare that there are no financial or personal relationships with other people or organizations that could inappropriately influence this study.

## References

1. Bell, S., Upchurch, P., Snavely, N., Bala, K.: Material recognition in the wild with the materials in context database. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3479–3487 (2015)
2. Caputo, B., Hayman, E., Mallikarjuna, P.: Class-specific material categorisation. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) IEEE, vol. 2, pp. 1597–1604 (2005)
3. Quan, Y., Xu, Y., Sun, Y., Luo, Y.: Lacunarity analysis on image patterns for texture classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 160–167 (2014)
4. Xue, J., Zhang, H., Dana, K.: Deep texture manifold for ground terrain recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 558–567 (2018)
5. Bruno, A., Collorec, R., Bézy-Wendling, J., Reuzé, P., Rolland, Y.: Texture analysis in medical imaging. In: Contemporary perspectives in three-dimensional biomedical imaging. IOS Press. pp. 133–164 (1997)
6. Cula, O.G., Dana, K.J.: Compact representation of bidirectional texture functions. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1. IEEE, vol. 1, pp. I–I (2001)
7. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. Int. J. Comput. Vis. **43**, 29 (2001)
8. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Int. J. Comput. Vis. **43**, 7 (2001)
9. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV, vol. 1. Prague. vol. 1, pp. 1–2 (2004)
10. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), vol. 2. IEEE., vol. 2, pp. 2169–2178 (2006)
11. Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3828–3836 (2015)
12. Hu, Y., Long, Z., AlRegib, G.:Multi-level texture encoding and representation (multer) based on deep neural networks. In: 2019 IEEE International Conference on Image Processing (ICIP) IEEE. pp. 4410–4414 (2019)
13. Liu, L., Chen, J., Fieguth, P., Zhao, G., Chellappa, R., Pietikäinen, M.: From BoW to CNN: two decades of texture representation for texture classification. Int. J. Comput. Vis. **127**, 74 (2019)
14. Basu, S., Mukhopadhyay, S., Karki, M., DiBiano, R., Ganguly, S., Nemani, R., Gayaka, S.: Deep neural networks for texture classification—a theoretical analysis. Neural Netw. **97**, 173 (2018)
15. Cavalin, P., Oliveira, L.S.: A review of texture classification methods and databases. In: 2017 30th SIBGRAPI Conference on graphics, patterns and images tutorials (SIBGRAPI-T) IEEE. pp. 1–8 (2017)
16. Liu, L., Fieguth, P., Guo, Y., Wang, X., Pietikäinen, M.: Local binary features for texture classification: taxonomy and experimental study. Pattern Recogn. **62**, 135 (2017)
17. Paul, R., Hawkins, S.H., Hall, L.O., Goldgof, D.B., Gillies, R.J.: Combining deep neural network and traditional image features to improve survival prediction accuracy for lung cancer patients from diagnostic CT. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC) IEEE. pp. 002,570–002,575 (2016)
18. Peeples, J., Xu, W., Zare, A.: Histogram layers for texture analysis. IEEE Trans. Artif. Intell. **3**(4), 541 (2021)
19. Zhang, H., Xue, J., Dana, K.: Deep ten: texture encoding network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 708–717 (2017)
20. Zhai, W., Cao, Y., Zhang, J., Zha, Z.J.: Deep multiple-attribute-perceived network for real-world texture recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3613–3622 (2019)
21. Yang, Z., Lai, S., Hong, X., Shi, Y., Cheng, Y., Qing, C.: DFAEN: double-order knowledge fusion and attentional encoding network for texture recognition. Expert Syst. Appl. **209**, 118223 (2022)
22. Bu, X., Wu, Y., Gao, Z., Jia, Y.: Deep convolutional network with locality and sparsity constraints for texture classification. Pattern Recogn. **91**, 34 (2019)

23. Basu, S., Karki, M., Mukhopadhyay, S., Ganguly, S., Nemani, R., DiBiano, R., Gayaka, S.: A theoretical analysis of deep neural networks for texture classification. In: 2016 International Joint Conference on Neural Networks (IJCNN) IEEE. pp. 992–999 (2016)
24. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. PMLR. pp. 2048–2057 (2015)
25. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3156–3164 (2017)
26. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
27. Luo, C., Zhan, J., Hao, T., Wang, L., Gao, W.: Shift-and-balance attention. arXiv preprint arXiv:2103.13080 (2021)
28. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
29. Gao, Z., Xie, J., Wang, Q., Li, P.: Global second-order pooling convolutional networks. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 3024–3033 (2019)
30. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
31. Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V.: Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 3286–3295 (2019)
32. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 510–519 (2019)
33. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R.: et al.: Resnest: split-attention networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2736–2746 (2022)
34. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.:Gcnet: non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE/CVF international conference on computer vision workshops (2019)
35. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.:ECA-Net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11534–11542 (2020)
36. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3606–3613 (2014)